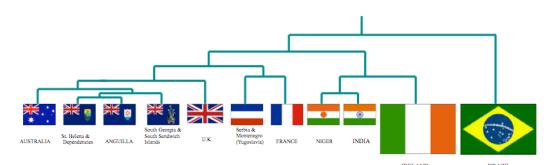


classification (and then some...) algorithms

Carlos Soares

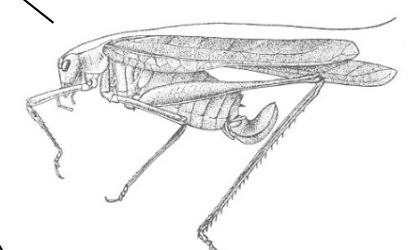
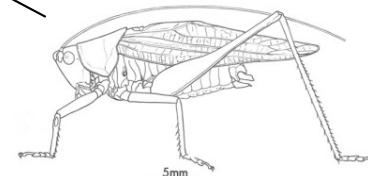
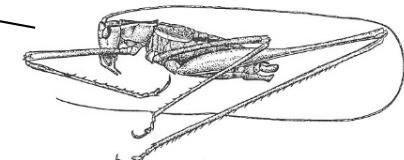
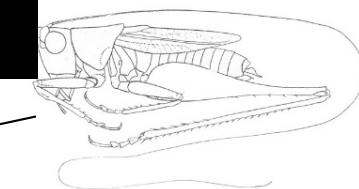
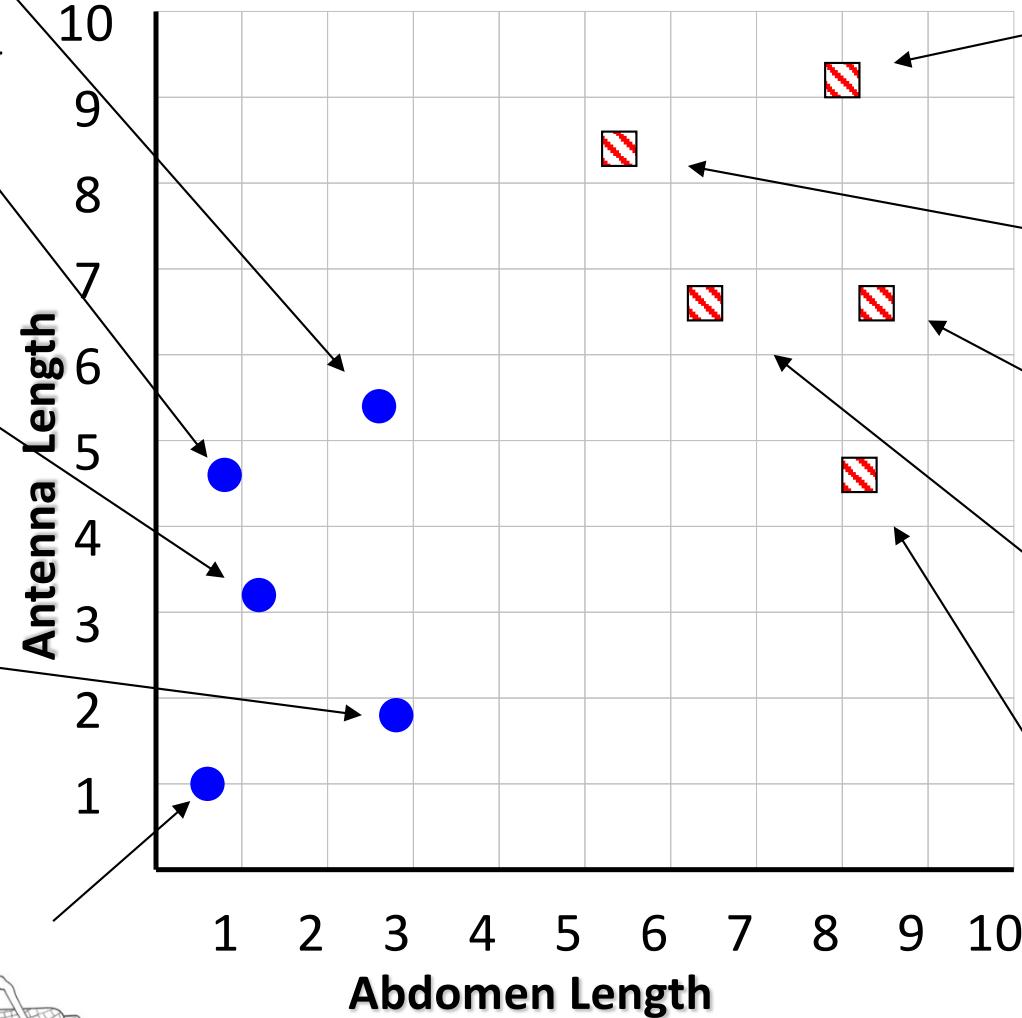
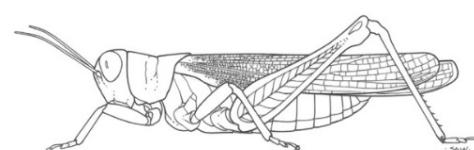
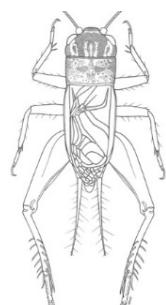
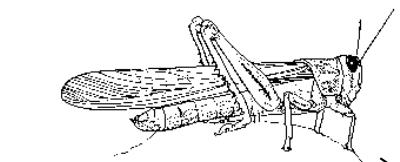
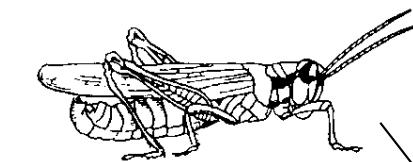
[based on materials kindly provided
by E. Keogh and J.M. Moreira]



plan

- geometric intuition of ML algorithms
 - Linear Classifiers
 - Nearest Neighbors
 - Decision Trees
 - Naive Bayes
 - Neural Networks
 - Support Vector Machines

running (hopping?) example



Grasshoppers

carlos soares - AC

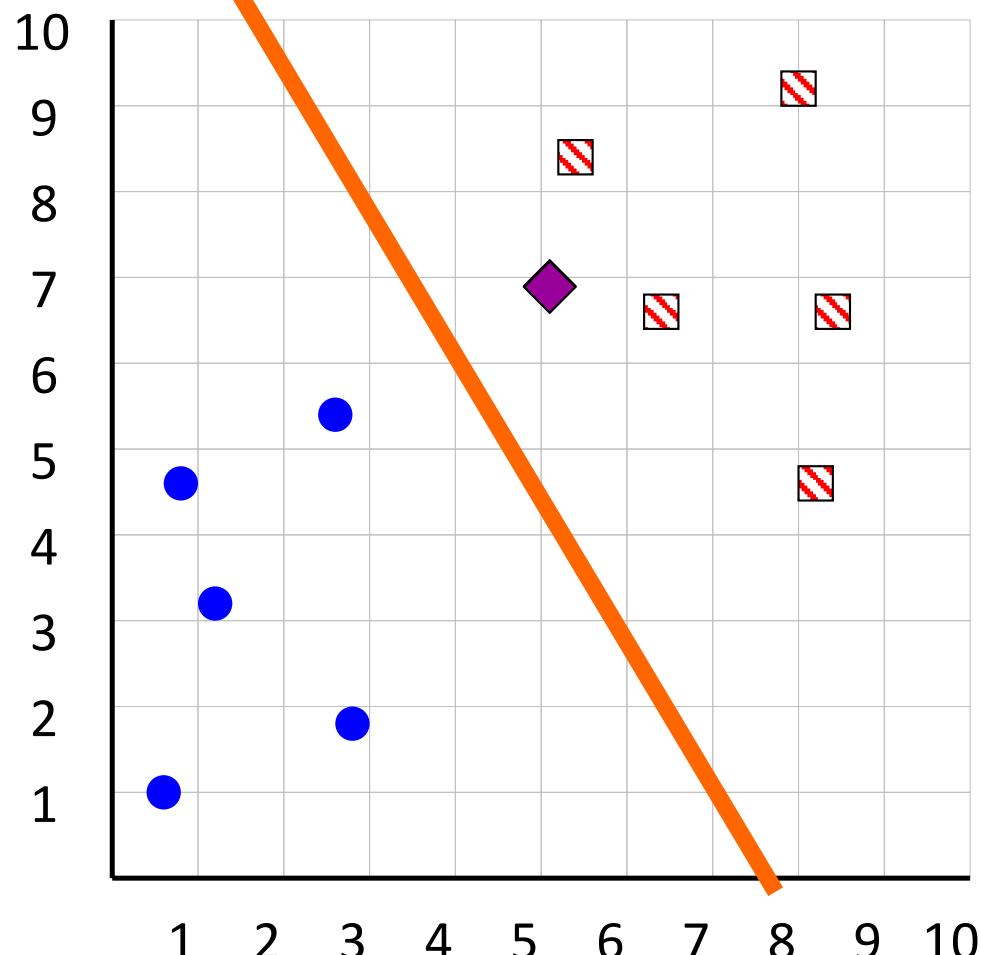
Katydids³

Simple Linear Classifier

RIA
D



R.A. Fisher
1890-1962



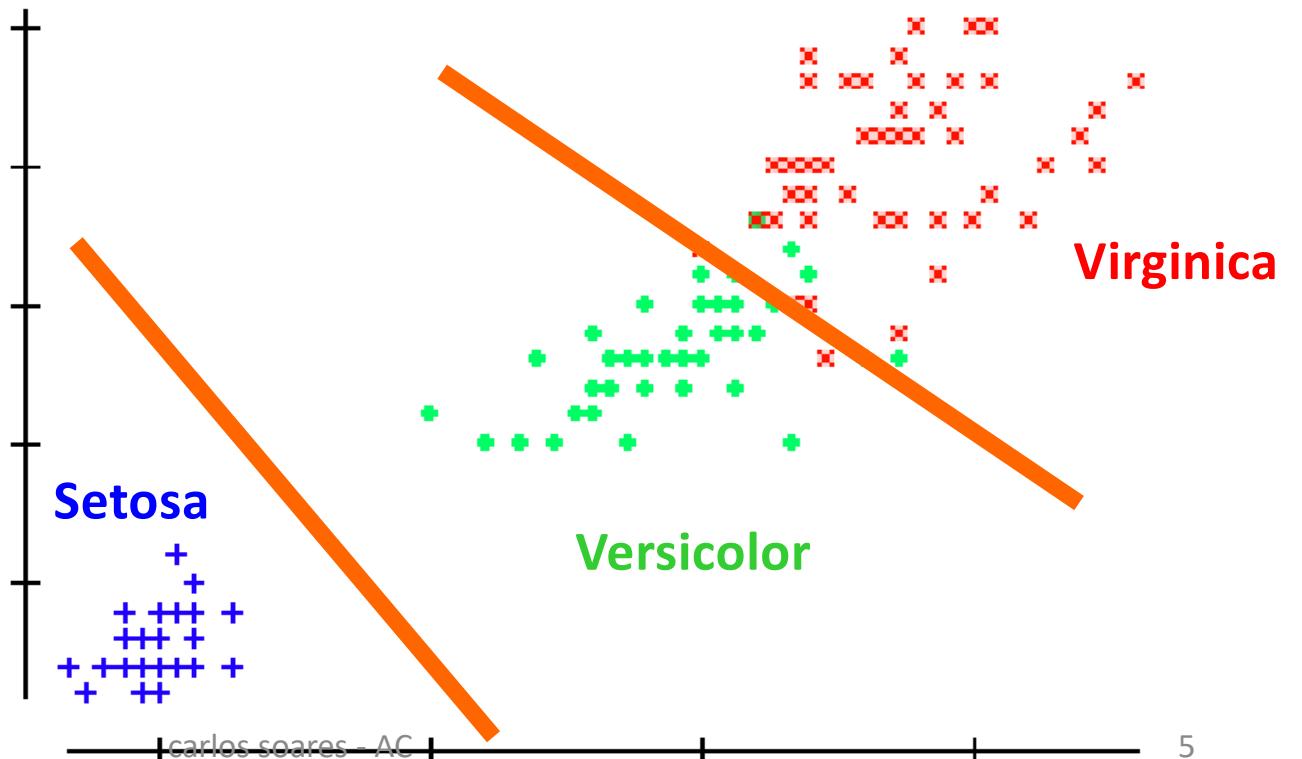
If previously unseen instance above the line
then
 class is **Katydid**
else
 class is **Grasshopper**

■ **Katydids**
● **Grasshoppers**

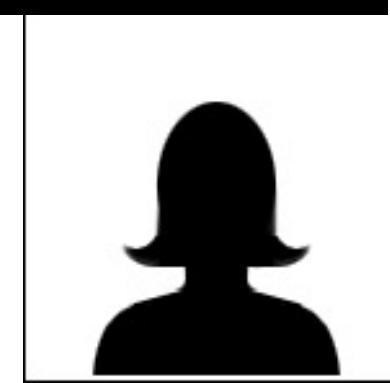
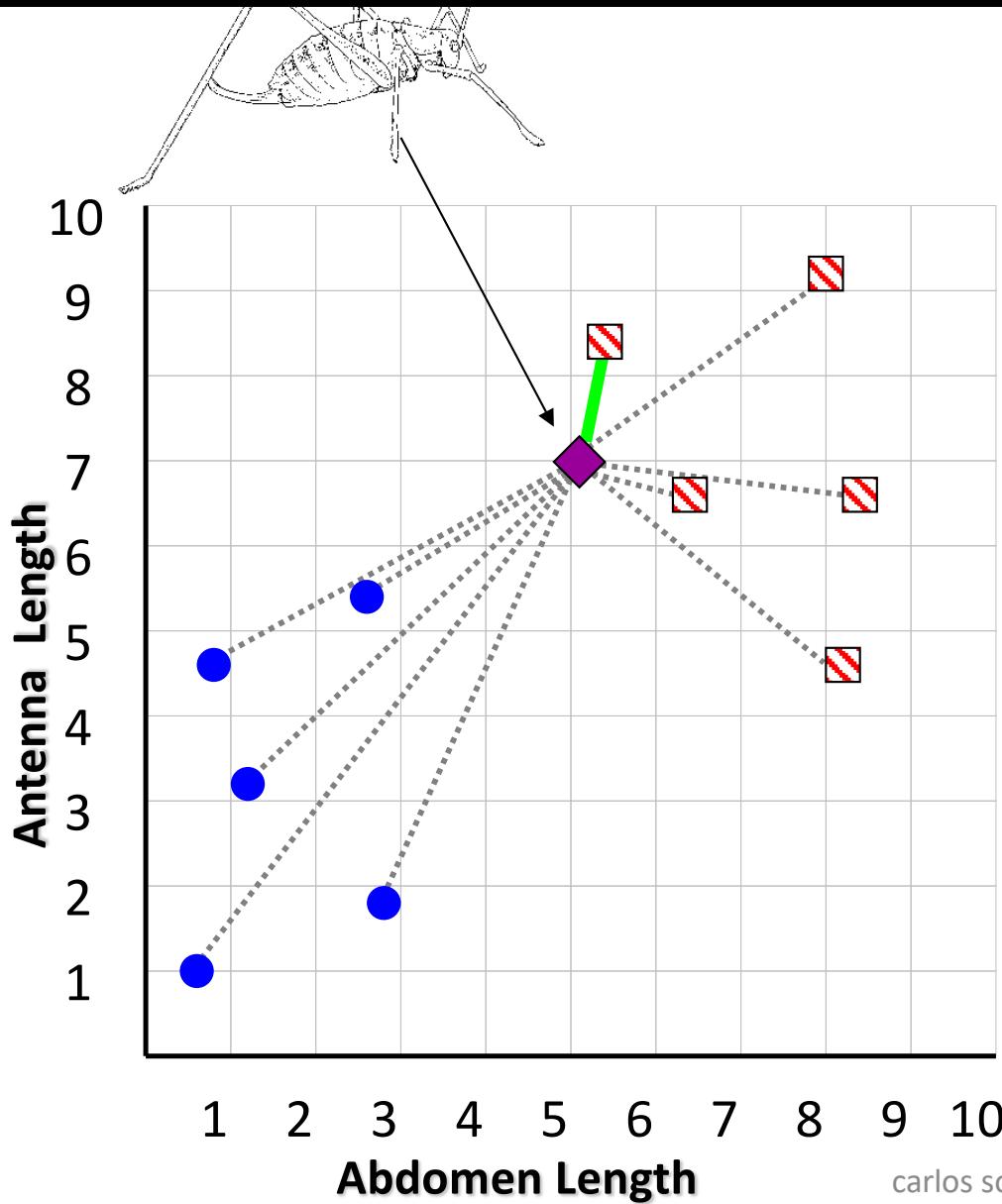
Linear Classifier for 3+ Classes

- linear classifier for N classes
- fit N-1 lines
 - Setosa vs Virginica/Versicolor
 - then
 - Virginica vs Versicolor.

If petal width > $3.272 - (0.325 * \text{petal length})$
then class = **Virginica**
Elseif petal width...



Nearest Neighbor Classifier



Evelyn Fix
1904-1965

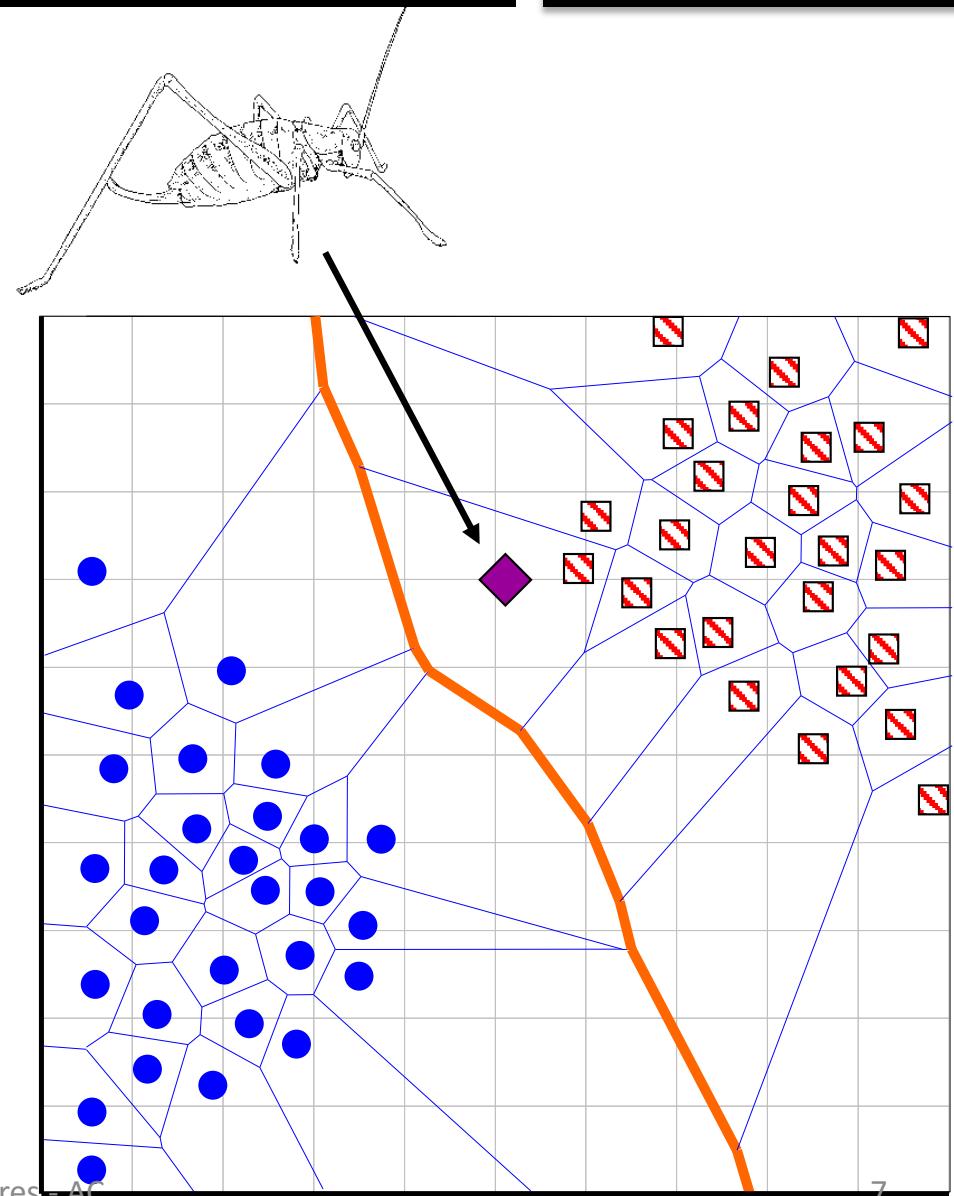
Joe Hodges
1922-2000

If the **nearest** instance to the **previously unseen instance** is a **Katydid**
class is **Katydid**
else
class is **Grasshopper**

■ **Katydid**
● **Grasshopper**

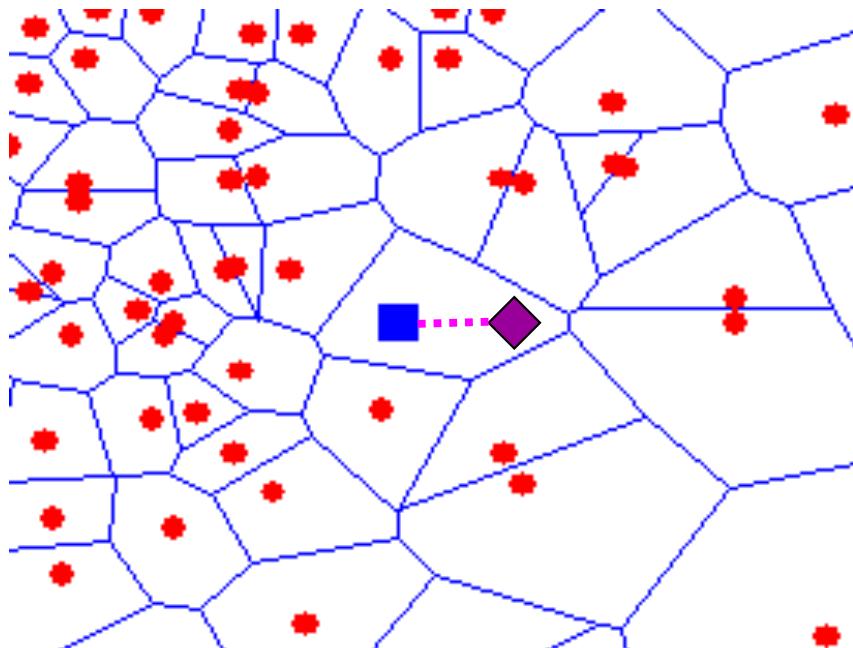
Geometrical Interpretation of Nearest Neighbor Classifier

- decision surface of a NN classifier
 - also called Dirichlet Tessellation,
 - ... Voronoi diagram
 - ... or Theissen regions
- implicit
 - never defined explicitly
 - divide the space into regions “belonging” to each instance
 - ... and corresponding class

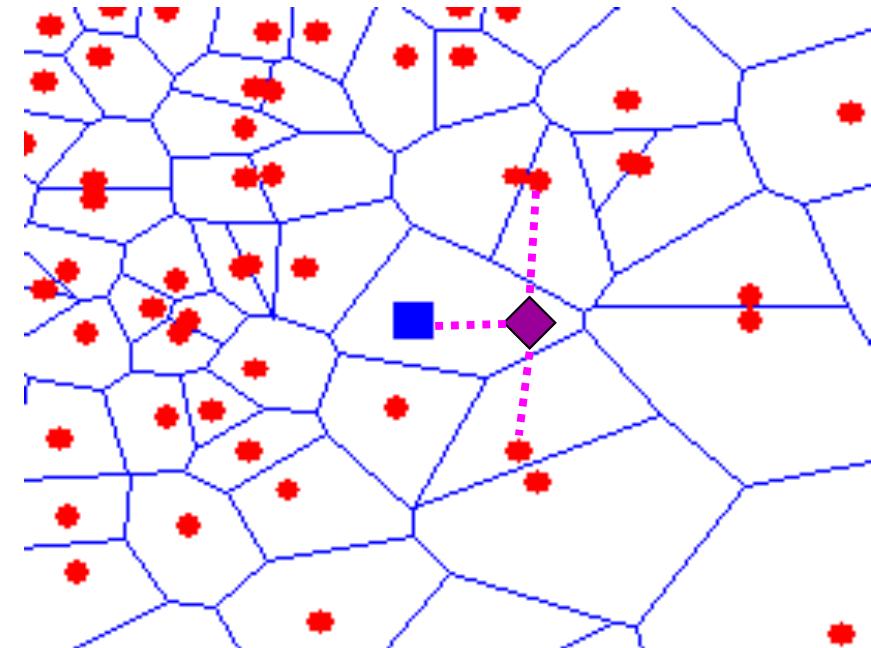


KNN Algorithm

- generalization of the nearest neighbor algorithm
 - find the nearest K instances
 - each one represents a vote
- K is typically chosen to be an odd number



$K = 1$



$K = 3$

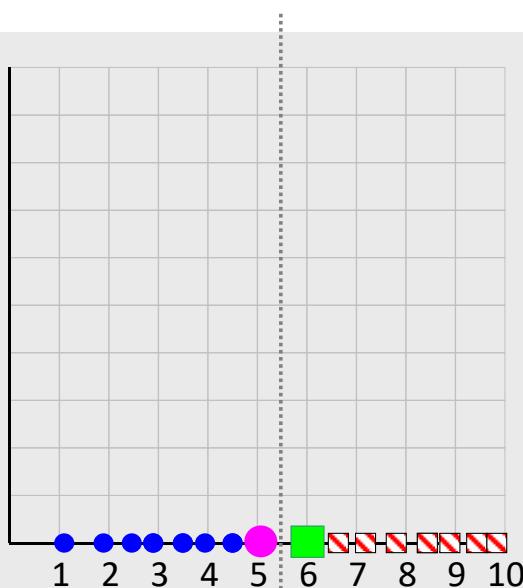
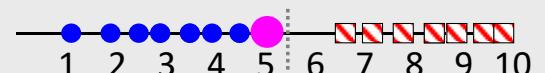
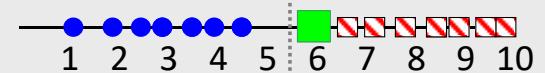
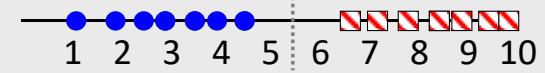
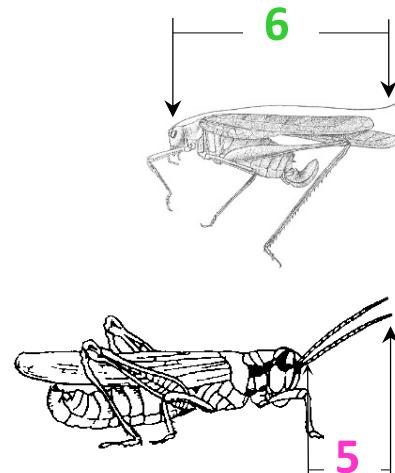
carlos soares - AC

8

Sensitivity to Irrelevant Attributes (1/2)

Suppose the following is true, if an insects antenna is longer than 5.5 it is a **Katydid**, otherwise it is a **Grasshopper**.

Using just the antenna length we get perfect classification!

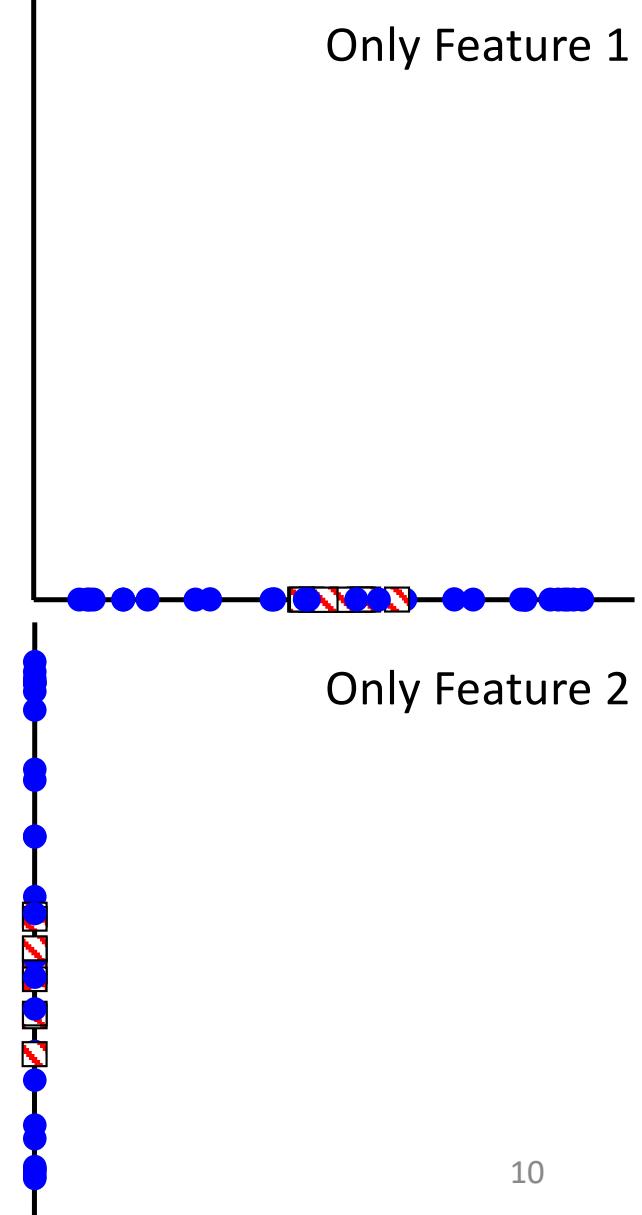
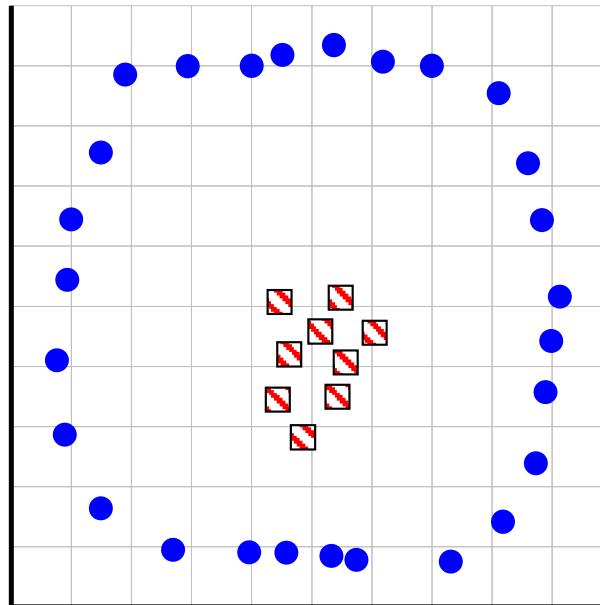


Suppose however, we add in an **irrelevant** feature, for example the insects mass.

Using both the antenna length and the insects mass with the 1-NN algorithm we get the wrong classification!

why search over feature subsets can fail

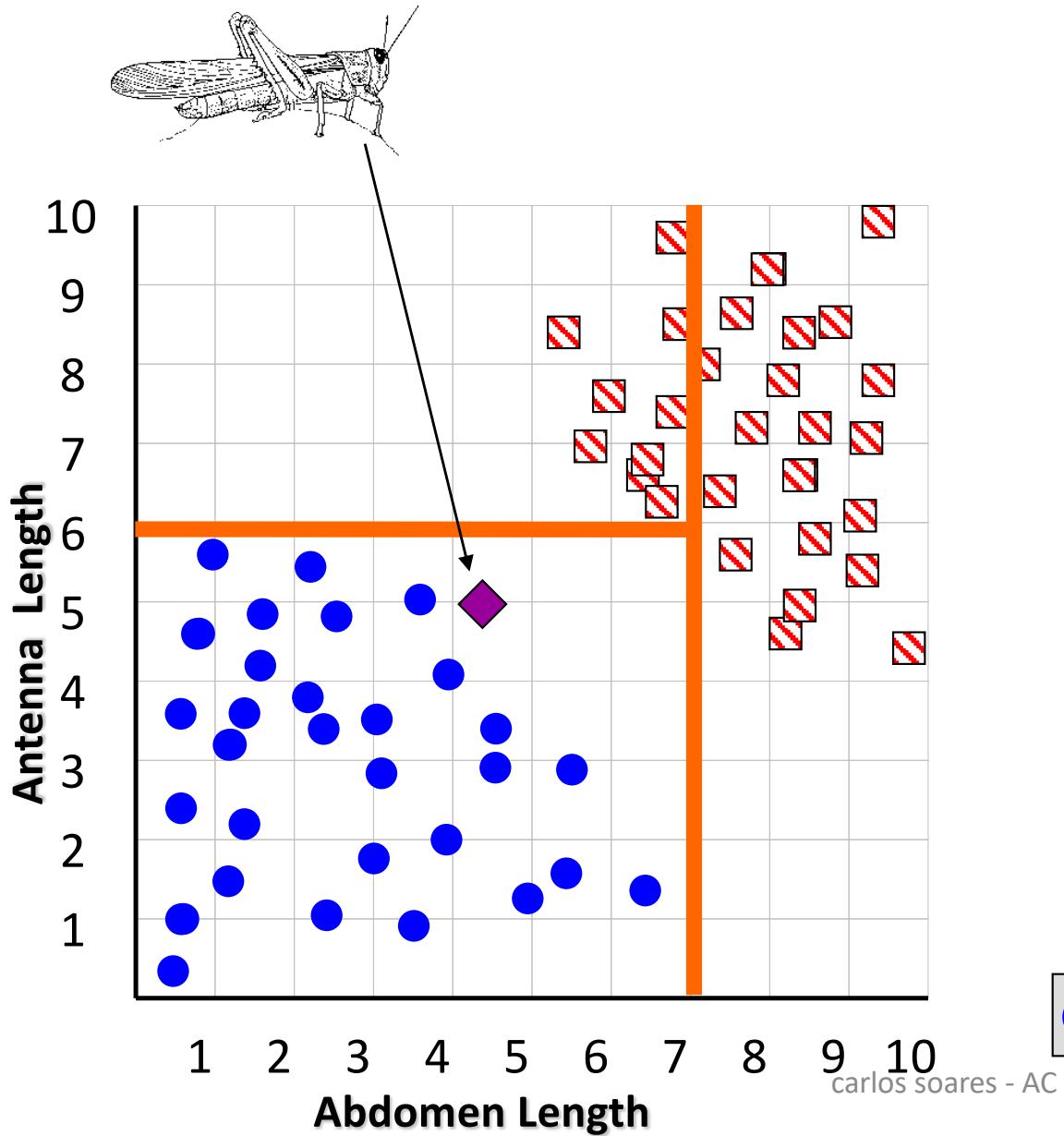
- problem
 - data with 100 features
 - Features 1 and 2 give perfect classification
 - ... remaining 98 are irrelevant
- poor results
 - 100 features
 - Feature 1 alone
 - Feature 2 alone
- only one subset gives good results
 - $2^{100} - 1$ possible subsets



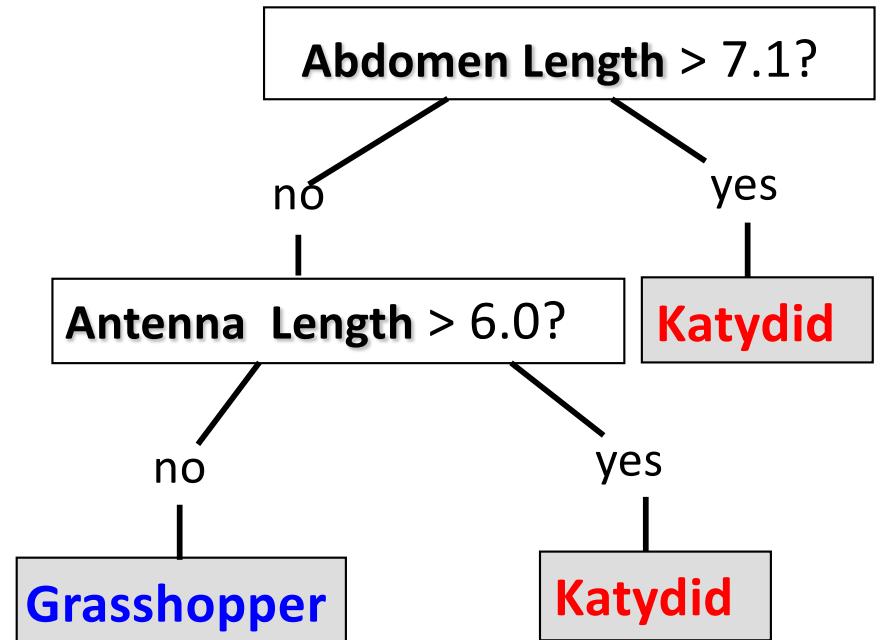
Advantages/Disadvantages of Nearest Neighbor

- Advantages
 - Simple to implement
 - Handles correlated features
 - Arbitrary class shapes
 - Defined for any distance measure
 - Handles streaming data trivially
- Disadvantages
 - Very sensitive to irrelevant features
 - Slow classification time for large datasets
 - Works best for real valued datasets

Decision Tree Classifier



Ross Quinlan



Avoid Overfitting in Classification

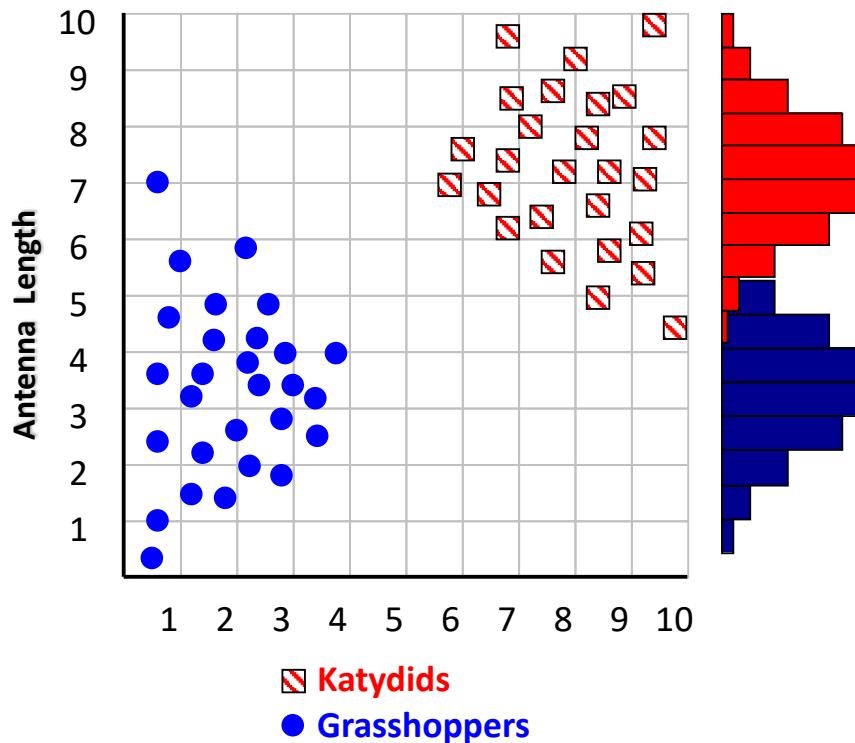
- Overfitting the training data
 - Too many branches
 - some may reflect anomalies due to noise or outliers
 - Poor accuracy for unseen samples
- Avoid overfitting
 - Prepruning
 - Halt tree construction early
 - do not split a node if this would result in the goodness measure falling below a threshold
 - ... difficult to choose an appropriate threshold
 - Postpruning
 - Get a sequence of progressively pruned trees
 - removing branches from a “fully grown” tree
 - Use a set of data different from the training data to decide which is the “best pruned tree”

Advantages/Disadvantages of Decision Trees

- Advantages
 - Easy to understand
 - Easy to generate rules
- Disadvantages
 - Overfitting
 - Does not handle correlated features very well
 - classifies by rectangular partitioning
 - Can be quite large
 - pruning is necessary

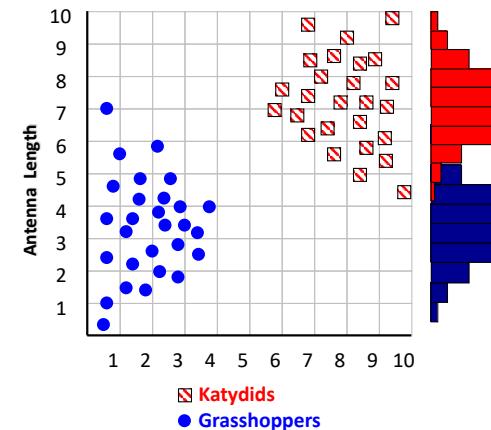
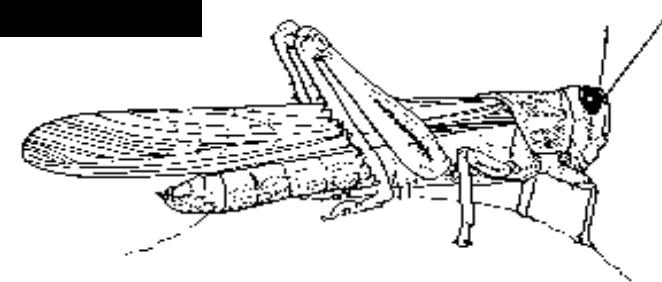
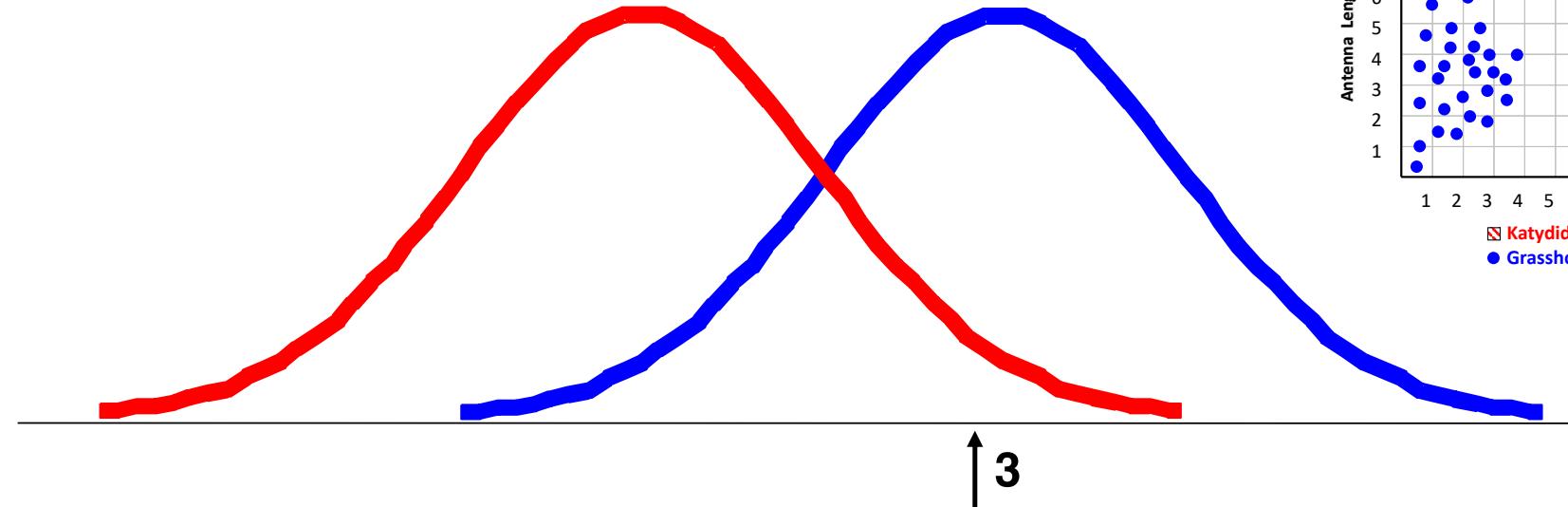
the distribution of bugs

- class histogram
 - the more data available
 - ... the more reliable



... can be used for prediction

- insect with 3 units long antennae
- ... more *probably* a **Grasshopper** or a **Katydid**?
 - given the distributions of antennae lengths we have seen



$p(c_j | d)$ = probability of class c_j , given that we have observed d

... but is it reliable?

how much evidence supports
 $p(c_j | d)$?

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Bayes classifies

- Bayesian classifiers use **Bayes theorem**, which says

$$p(c_j | d) = \frac{p(d | c_j) p(c_j)}{p(d)}$$

– $p(c_j | d)$ = probability of instance d being in class c_j ,

This is what we are trying to compute

– $p(d | c_j)$ = probability of generating instance d given class c_j ,

We can imagine that being in class c_j , causes you to have feature d with some probability

– $p(c_j)$ = probability of occurrence of class c_j ,

This is just how frequent the class c_j , is in our database

– $p(d)$ = probability of instance d occurring

This can actually be ignored, since it is the same for all classes

... assuming independence

if attributes have independent distributions

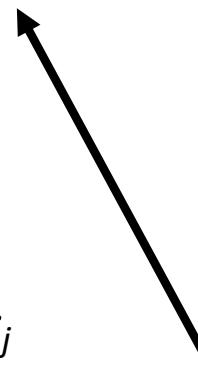
$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \dots * p(d_n|c_j)$$



The probability of class c_j generating instance d , equals....



The probability of class c_j generating the observed value for feature 1, multiplied by..



The probability of class c_j generating the observed value for feature 2, multiplied by..

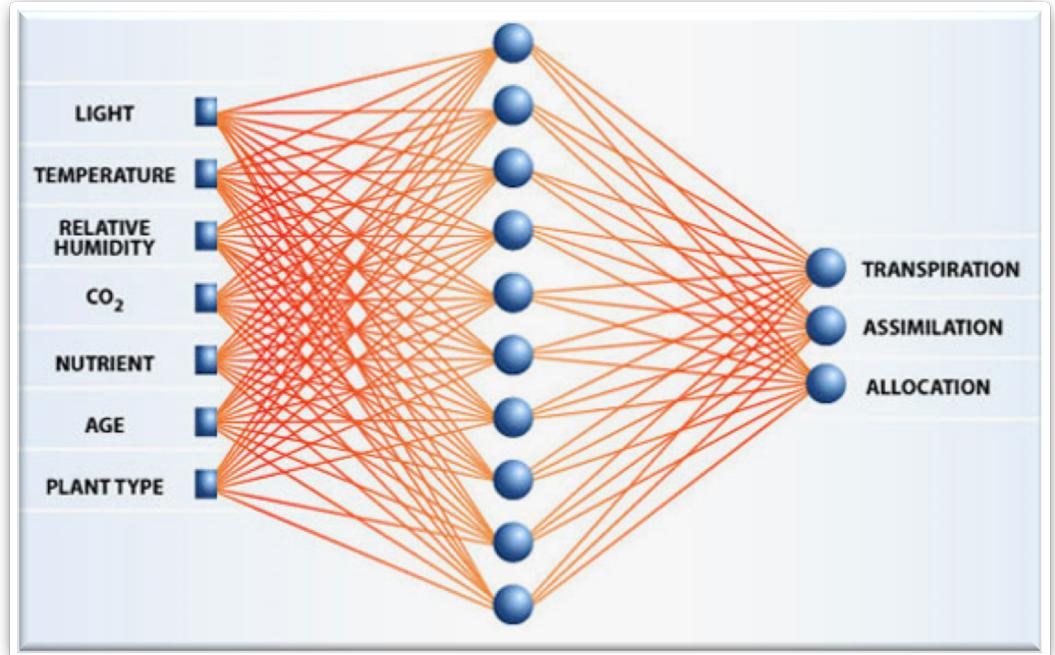


Advantages/Disadvantages of Naive Bayes

- Advantages
 - Fast
 - to train
 - single scan
 - to classify
 - Not sensitive to irrelevant features
 - Handles real and discrete data
 - Handles streaming data well
- Disadvantages
 - Assumes independence of features

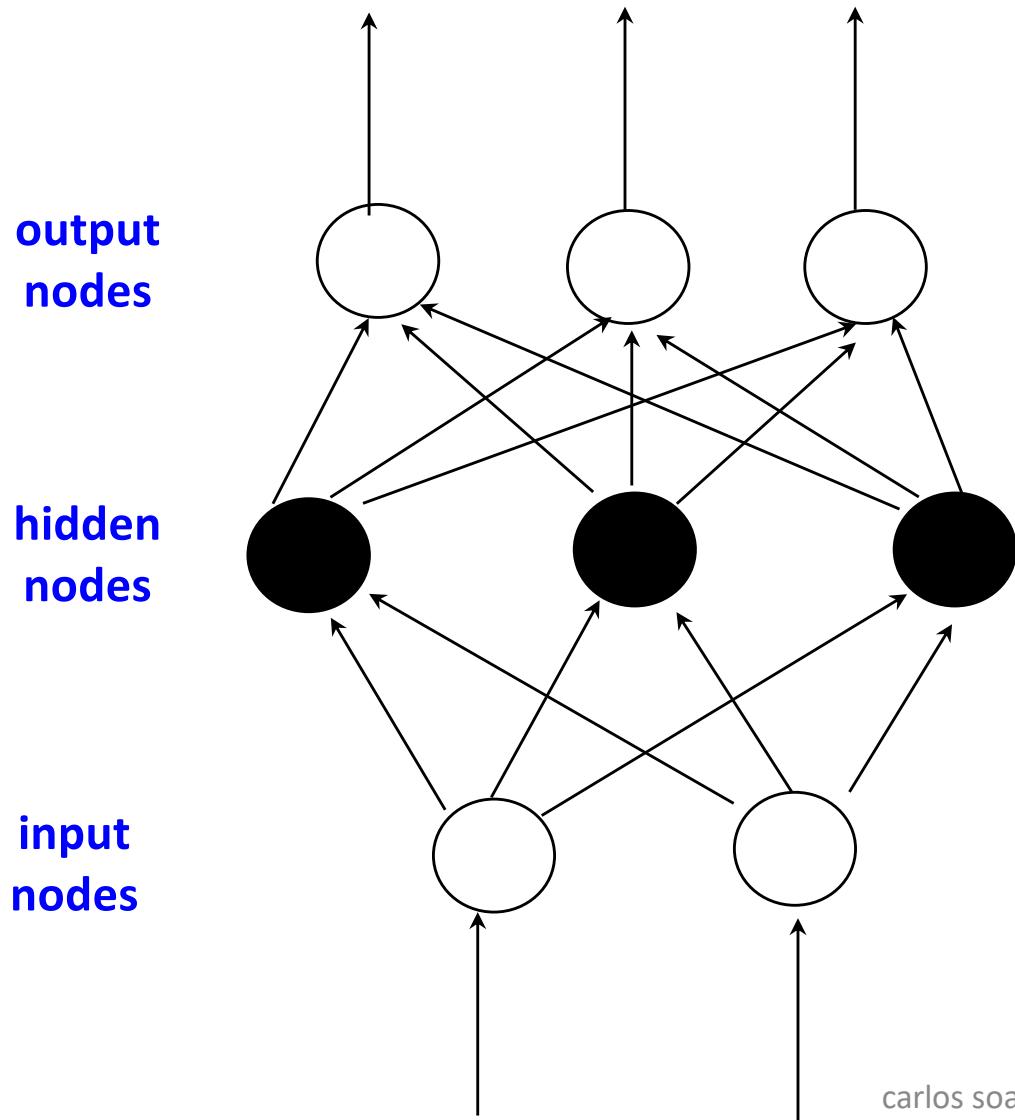
neural network learning

- set of neurons
 - input/output units
- connected
- ... with weights
- (supervised) learning
 - adjust weights to ensure outputs to given inputs are the expected ones
- predicting
 - feed input values
 - collect outputs



http://aemc.jpl.nasa.gov/activities/bio_regen.cfm

summary: feedforward



$$\hat{y}_k = \frac{1}{1 + e^{-w_{0k} + \sum_{i \in \{\text{all units feeding into unit } k\}} w_{ik} y_i}}$$

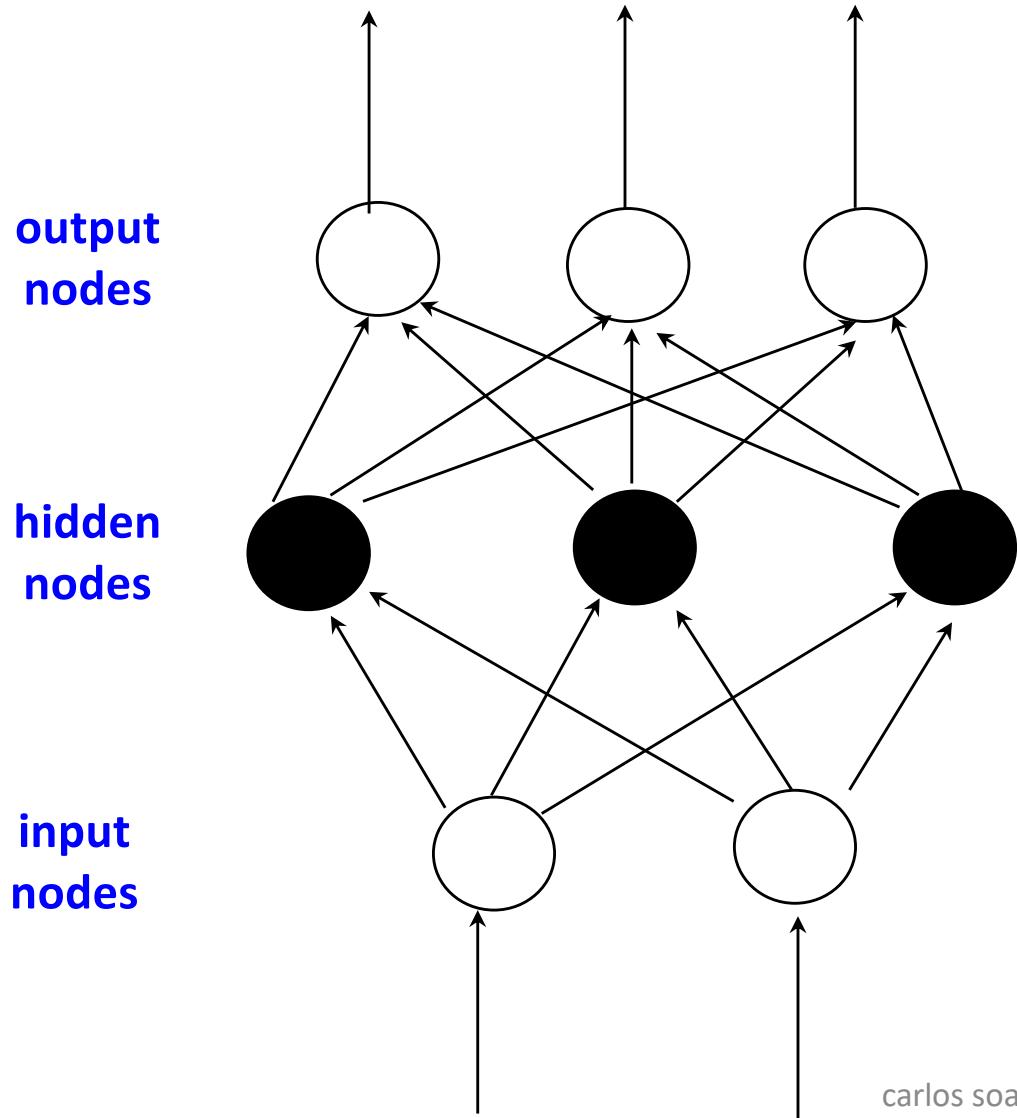
$w_{j,k}$ weights

$$y_i = \frac{1}{1 + e^{-w_{0i} + \sum_{j \in \{\text{all units feeding into unit } i\}} w_{ji} x_j}}$$

$w_{i,j}$ weights

x_i input values

summary: backprop



$$\delta_k = \hat{y}_k(1 - \hat{y}_k)(\hat{y}_k - y_k)$$

$$w_{j,k}^{new} = w_{j,k} + r \cdot y_j \cdot \delta_k$$

$$\delta_j = y_j(1 - y_j) \sum_{k \in \{\text{all units fed by this unit}\}} \delta_k$$

$$w_{i,j}^{new} = w_{i,j} + r \cdot x_i \cdot \delta_j$$

discussion



- universal
 - fit any continuous function
- versatile
 - output may be one or more discrete and real values
- online
 - application and learning are intertwined
- robust
 - errors and noisy data
- fast
 - ... application to new examples
- parallel



- slow
 - ... training
- low usability
 - empirical parameter tuning
 - network topology and learning rate
- low interpretability
 - understand the weights
- low adaptability
 - not easy to incorporate domain knowledge

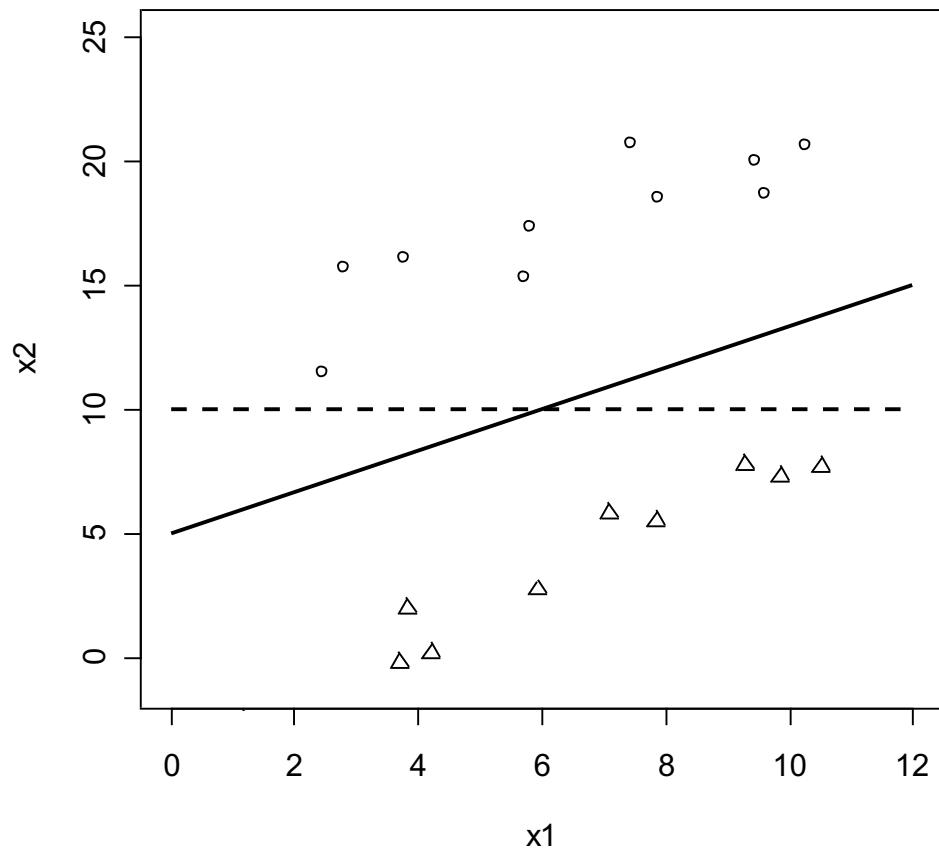
universal approximator: the whole story

- mlps are a class of universal approximators
 - 1 hidden layer
- so what's the catch?
 - provided sufficiently many hidden units...

overview

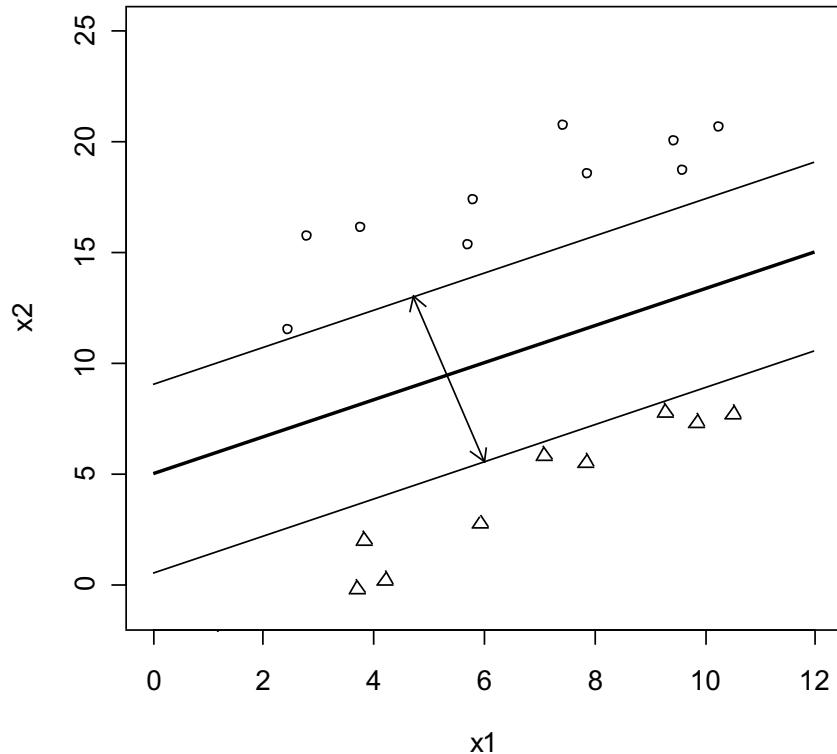
- Support Vector Machines (SVM)
 - linear learning machines with maximization of margin
 - **better separation between classes**
 - duality
 - **higher robustness to the curse of dimensionality**
 - kernel trick
 - **non-linear models**
- according to Bennet & Campbell, “Support Vector Machines: Hype of Hallelujah?”, SIGKDD Explorations, 2000
 - geometric intuition
 - elegant math
 - theoretical guarantees
 - practical algorithms

best separating hyperplane?



margin maximization: intuition

- margin, γ :
 - sum of the shortest distances between points of each class and the separating hyperplane
- margin maximization
 - hyperplane “farthest away” from both classes simultaneously
- less risk of overfitting!



margin maximization: formal definition

- it can be proved that

$$\gamma = \frac{1}{\|\mathbf{w}\|_2}$$

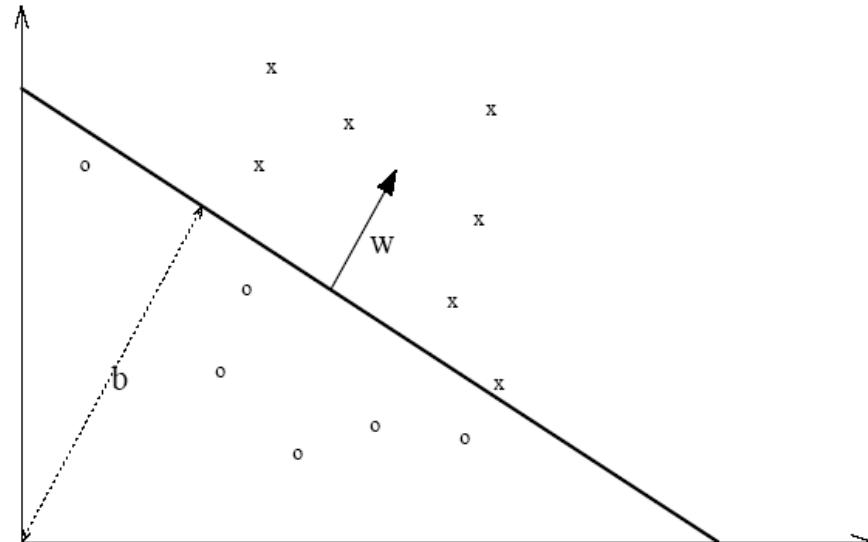
- ... thus, maximizing the margin is the same as

minimize

$$\|\mathbf{w}\|_2$$

subject to $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1$
the constraints $i = 1, \dots, l$

- quadratic programming problem
 - i.e. a lot of work in the area of optimization in this problem can be reused, most importantly...



fonte: www.kernel-machines.org

... duality

- dual problem

$$\text{maximize} \quad \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

subject to
the constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0 \\ \alpha_i &\geq 0, i = 1, \dots, l \end{aligned}$$

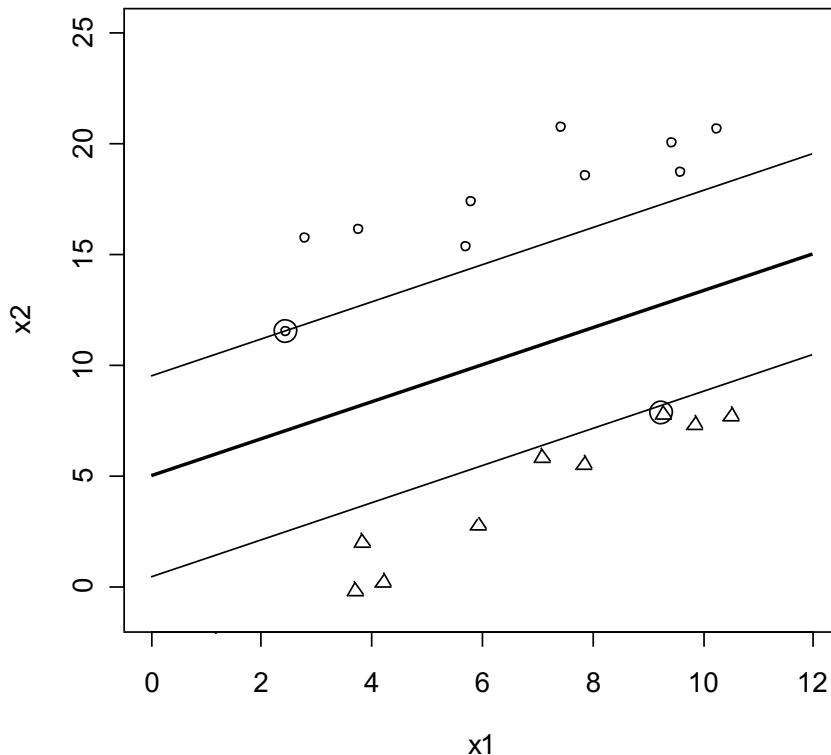
- linear prediction problem

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x}_{new} \rangle + b$$

model

- subset of examples that determine the margin
 - frontier
 - other points are irrelevant

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x}_{new} \rangle + b$$



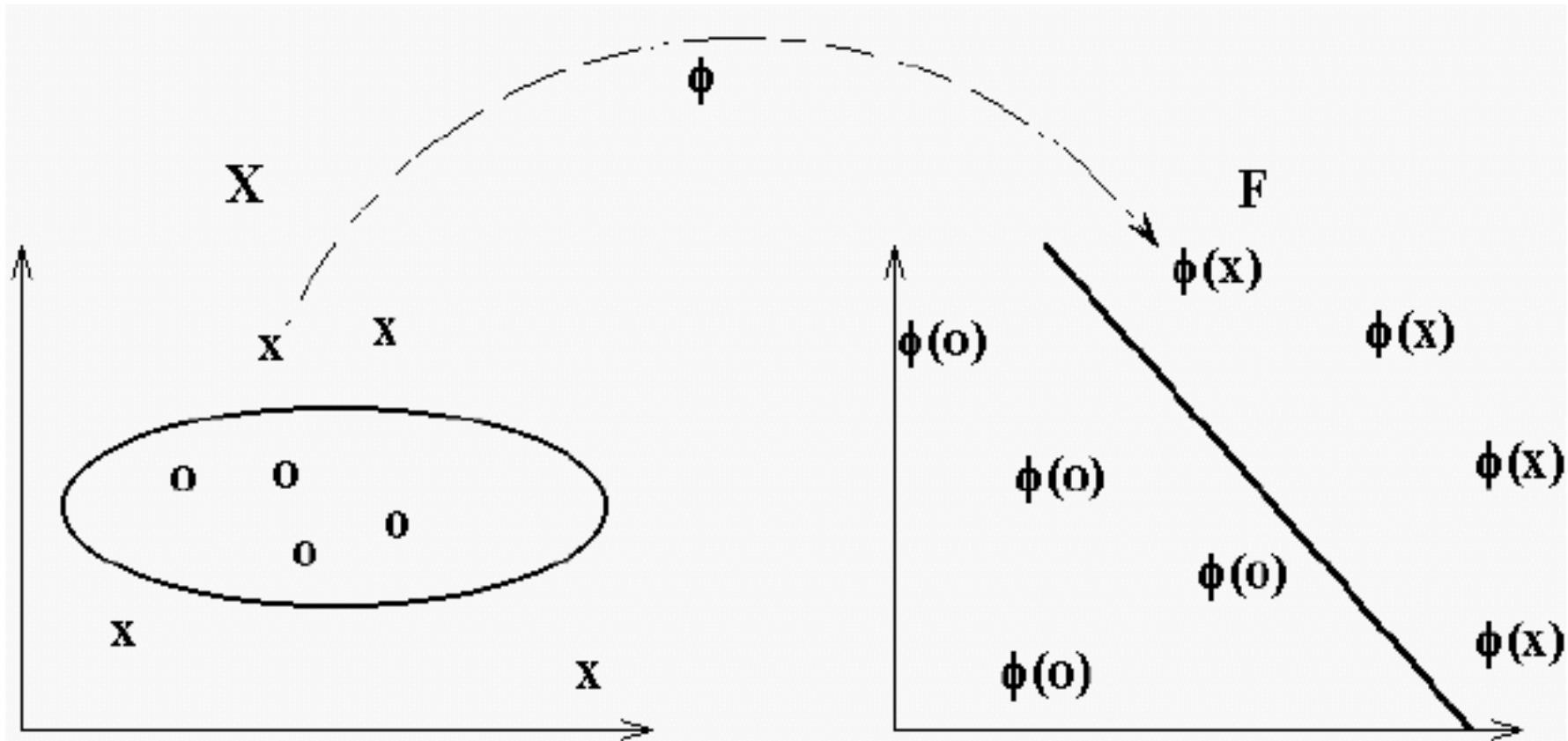
pros

- QP methods are very mature
- statistical learning theory
 - bounds to the generalization error based on the training error
- results independent of initial conditions
 - order of presentation of examples
 - initializations
- convex problem
 - no local minima
 - ... reducing the probability of overfitting
- dual is independent of number of attributes
 - minimizing effect of the curse of dimensionality

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

non-linear functions

- map attributes to space where linear discrimination is possible



fonte: www.kernel-machines.org

but, which space?

- “A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in low-dimensional space”
 - Cover (95)
- i.e., the bigger, the better... or maybe not
 - theoretical challenge: curse of dimensionality
 - practical challenge: computational resources
 - e.g. memory

dual problem in space...

- dual problem

$$\text{maximize} \quad \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \phi(x_i) \cdot \phi(x_j) \rangle$$

subject to
the constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0 \\ \alpha_i &\geq 0, i = 1, \dots, l \end{aligned}$$

- (non-)linear prediction function

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i \langle \phi(x_i) \cdot \phi(\mathbf{x}_{new}) \rangle + b$$

the kernel trick

- kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$
- ... integrated in the previous method

maximize $\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$

subject to
the constraints $\sum_{i=1}^l y_i \alpha_i = 0$
 $\alpha_i \geq 0, i = 1, \dots, l$

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_{new}) + b$$

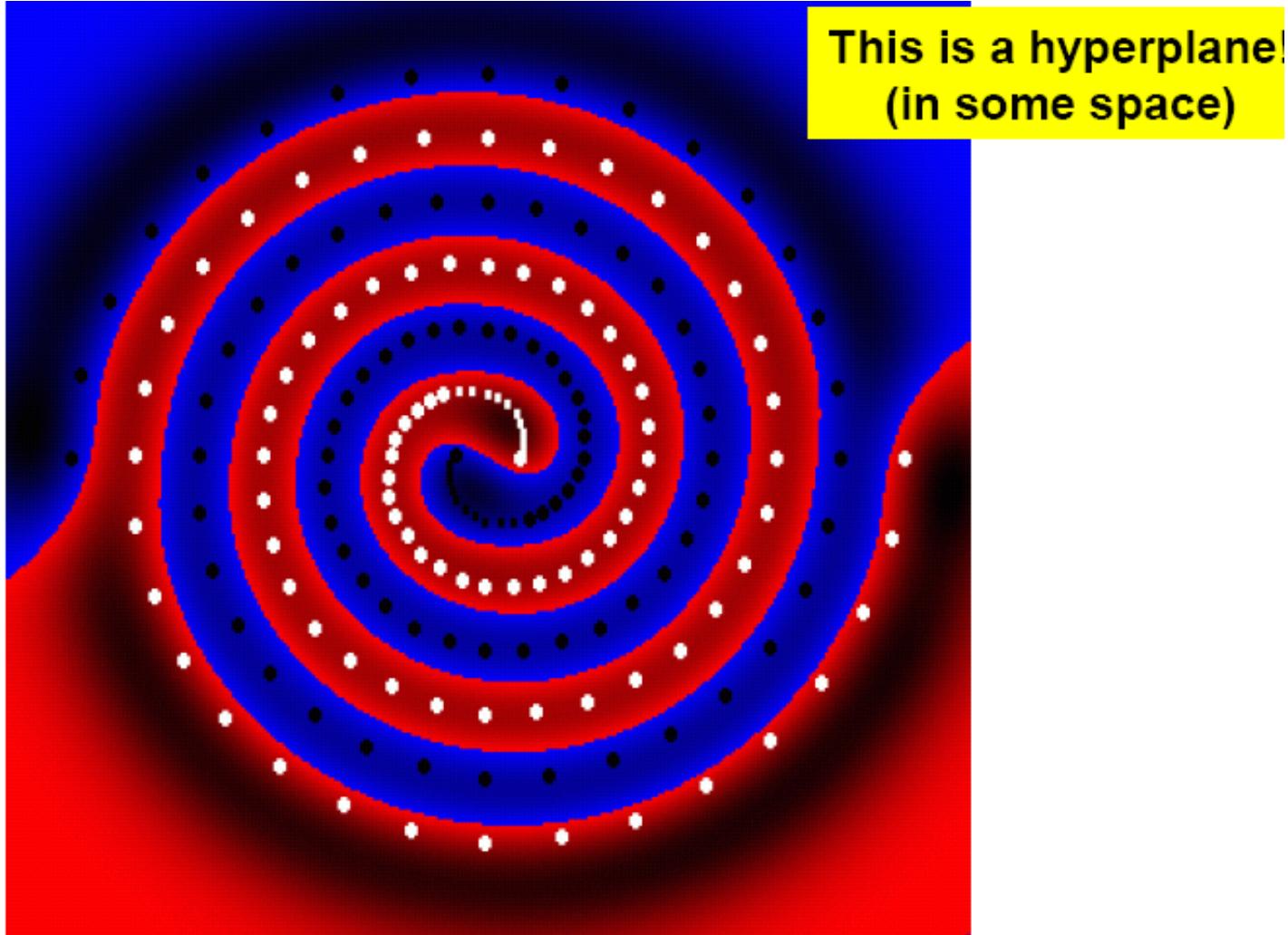
kernels + maximization of the margin with the dual

- pros: kernel
 - problem projected to higher dimension space...
 - not necessarily...
 - but potentially infinite...
 - implicit
 - no need to compute $\phi(x)$...
 - ... which may be unknown!

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$$

- pros: margin maximization
 - dual doesn't depend on the number of variables
 - minimize the effect of the curse of dimensionality
 - problem remains convex
 - unique solution

gaussian kernel



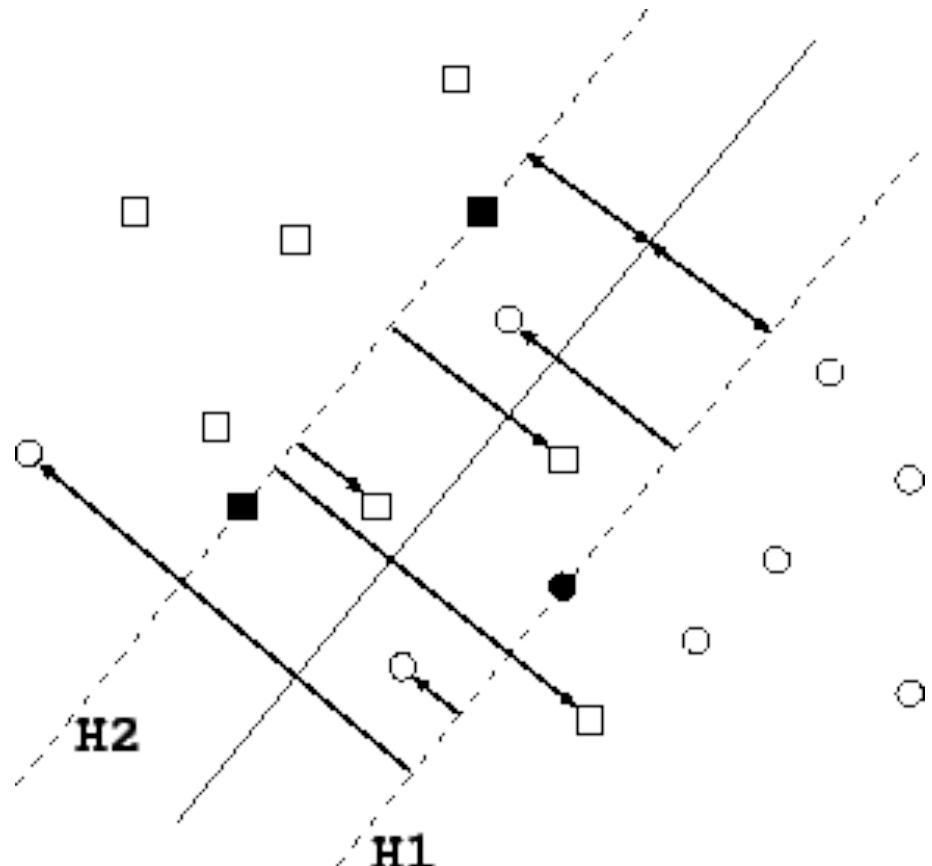
$$e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

relax the objective function

- soft margin
 - maximize the margin
 - minimize error

$$\frac{1}{2} \|\mathbf{w}\|_2 + C \sum_{i=1}^l z_i$$

- C , regularization constant
 - compromise between the importance of the margin and the error
 - yet another parameter...
- most common type of SVM



SVM Soft Margin: dual problem

maximize $\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$

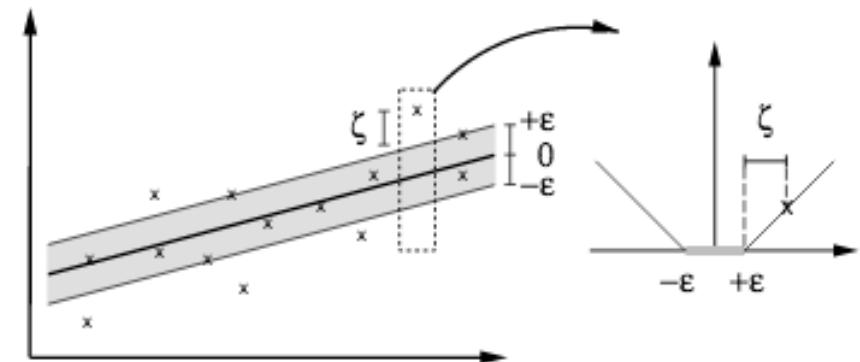
subject to
the constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0 \\ C \geq \alpha_i &\geq 0, i = 1, \dots, l \end{aligned}$$

- only one difference
 - weight of each example is limited

SVM for Regression

- margin
 - minimize the tube “around” the data
 - Instead of maximizing the distance to closest examples from each class



source: <http://alex.smola.org/papers/2003/SmoSch03b.pdf>